

**Vysoká škola báňská – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra telekomunikační techniky**

**Sběr dat ze sensorů připojených k mikrokontroléru Arduino
a následný přenos dat do počítače s OS Linux**

**Data Collection from Sensors Connected to Microcontroller
Arduino and Data Transmission to Linux Computer**

2019

Zdeněk Kalich

Zadání bakalářské práce

Student: **Zdeněk Kalich**

Studijní program: B2647 Informační a komunikační technologie

Studijní obor: 2601R013 Telekomunikační technika

Téma: Sběr dat ze sensorů připojených k mikrokontroleru Arduino a následný
přenos dat do počítače s OS Linux
Data Collection from Sensors Connected to Microcontroller Arduino and
Data Transmission to Linux Computer

Jazyk vypracování: čeština

Zásady pro vypracování:

Cílem bakalářské práce je zajistit sběr dat z různých sensorů, připojených k mikrokontroléru Arduino a jejich následný přenos do počítače s OS Linux.

Řešení práce spočívá ve splnění následujících úkolů:

1. Studium a popis různých typů sensorů, připojitelných k Arduino.
2. Studium a popis komunikačních protokolů pro přenos dat.
3. Návrh řešení automatizovaného sběru dat pomocí sensorů.
4. Přenos naměřených dat do počítače a jejich zpracování.

Seznam doporučené odborné literatury:

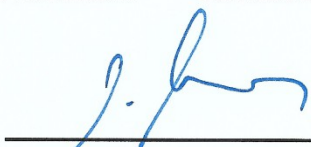
[1] Karvinen T., Karvinen K., Valtokari V. *Make: Sensors: A Hands-On Primer for Monitoring the Real World with Arduino and Raspberry Pi*. Maker Media 2014

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.


Vedoucí bakalářské práce: **Ing. Pavel Nevlud**

Datum zadání: 01.09.2018

Datum odevzdání: 30.04.2019


prof. Ing. Miroslav Vozňák, Ph.D.
vedoucí katedry





prof. Ing. Pavel Brandštetter, CSc.
děkan fakulty

Prohlášení studenta

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě dne: 29. dubna 2019


.....
podpis studenta

Poděkování

Rád bych poděkoval vedoucímu bakalářské práce panu Ing. Pavlu Nevludovi za odbornou pomoc a konzultaci při vytváření této bakalářské práce.

Abstrakt

Tato bakalářská práce se zabývá návrhem automatizovaného sběru dat ze senzorů připojených k mikrokontroléru Arduino a následným přenosem dat do počítače s OS Linux. V teoretické části popisuje, co je to senzor a různé typy senzoru, které lze připojit k mikrokontroléru Arduino. Dále popisuje komunikační protokoly pro přenos dat. V praktické části se zabývá návrhem a realizací daného zařízení. Pro zpracování a ukládání dat do databáze InfluxDB je použit skript v Pythonu. K vizualizaci naměřených dat je použit software Grafana.

Klíčová slova

Arduino; senzor; I2C; InfluxDB; Grafana; Bluetooth; Python

Abstract

This bachelor thesis deals with the design of automated data collection from sensors connected to Arduino microcontroller and subsequent data transfer to a Linux computer. In the theoretical part it describes the sensor and various types of sensor that can be connected to the Arduino microcontroller. It also describes communication protocols for data transmission. The practical part deals with the design and implementation of the device. Python script is used to process and store data in InfluxDB. The Grafana software is used to visualize the measured data.

Key words

Arduino; sensor; I2C; InfluxDB; Grafana; Bluetooth; Python

Seznam použitých zkratek

Zkratka	Význam
I2C	Inter Integrated Circuit
LED	Light Emitting Diode
SPI	Serial Peripheral Interface
UART	Universal Asynchronous Receiver Transmitter
WiFi	Wireless Fidelity
USB	Universal Serial Bus
GPS	Global Positioning System
ISM	Industrial, Scientific and Medical
OS	Operating System
UV	Ultraviolet

Obsah

Úvod.....	- 9 -
1 Senzory.....	- 10 -
1.1 Senzory pro měření vzdálenosti	- 10 -
1.2 Senzory teploty, vlhkosti a barometrického tlaku	- 10 -
1.3 Senzory pohybu.....	- 11 -
1.4 Senzory hluku.....	- 11 -
1.5 Senzory světla a barev	- 11 -
1.6 Senzory plynů, kouře a kvality ovzduší	- 12 -
1.7 Proudové senzory	- 13 -
1.8 Gyroskopy a akcelerometry.....	- 13 -
2 Komunikační protokoly.....	- 14 -
2.1 UART	- 14 -
2.2 SPI.....	- 15 -
2.3 I2C.....	- 16 -
2.4 USB	- 19 -
2.5 Bluetooth	- 20 -
2.5.1 Bluetooth profily	- 21 -
3 Návrh řešení automatizovaného sběru dat pomocí senzorů	- 22 -
3.1 Návrh řešení	- 22 -
3.2 Kontrola senzoru intenzity světla BH1750.....	- 23 -
3.3 Kontrola senzoru BME280.....	- 24 -
3.4 Kontrola senzoru INA3221	- 26 -
3.5 Kontrola Bluetooth modulu HC-05	- 28 -
3.6 Propojení všech zařízení.....	- 31 -
3.7 Vkládání naměřených dat do databáze	- 33 -
4 Vizualizace naměřených dat.....	- 36 -
Závěr	- 39 -
Použitá literatura	- 40 -
Obsah elektronické přílohy	- 42 -

Úvod

Senzor je zařízení, které se využívá v běžném životě pro měření nejrůznějších fyzikálních veličin a zaznamenávání událostí. Existuje nepřeberné množství senzorů, které můžeme využít. Sensory mohou být různě velké, přesné a mohou využívat různé komunikační protokoly pro přenos naměřených dat. Mají také různou pořizovací cenu s ohledem na kvalitu zpracování a přesnost měření.

Tato bakalářská práce se zabývá popisem základních typu senzoru, které lze připojit k mikrokontroléru Arduino. Sensory jsou rozděleny a popsány v několika různých kategoriích.

Druhá kapitola se zabývá popisem komunikačních protokolů pro přenos dat. Jsou zde popsány ty nejpoužívanější protokoly pro přenos dat mezi senzorem a mikrokontrolérem Arduino. Dále jsou zde popsány protokoly, které slouží pro přenos dat z mikrokontroléru Arduino do počítače.

Třetí kapitola popisuje návrh a realizaci automatizovaného sběru dat ze senzoru připojených k mikrokontroléru Arduino do počítače s OS Linux. Počítač s OS Linux je realizován jako virtuální počítač na fyzickém počítači s OS Windows. Nejprve jsou samostatně otestována jednotlivá zařízení, která budou použita při celkové realizaci a poté následuje propojení všech zařízení a popis způsobu přenosu naměřených dat pomocí Bluetooth modulu. Pro ukládání naměřených dat byla zvolena databáze časových řad InfluxDB. Pro zpracování a ukládání přicházejících naměřených dat na sériový port počítače do databáze InfluxDB je použit skript v Pythonu.

Poslední část práce se zabývá propojením databáze s vizualizačním softwarem Grafana a vizualizací naměřených dat v tomto softwaru.

1 Senzory

Senzor je zařízení, které slouží pro detekování události nebo změny ve svém okolí. Vstupem může být teplo, světlo, vlhkost, hluk, pohyb nebo jakýkoliv v řady jevů. Senzory využíváme v každodenním životě a každý z nás se s nimi již určitě setkal, ať už jde o senzory pro automatické otevírání dveří, vrat či rozsvícení světel. Senzory mohou mít různou velikost, přesnost, cenu a mohou využívat rozdílné fyzikální principy pro měření dané veličiny.

1.1 Senzory pro měření vzdálenosti

Senzory pro měření vzdálenosti můžeme rozdělit do dvou kategorií, na ultrazvukové a infračervené. Princip ultrazvukových senzorů je podobný sonaru nebo radaru. Hodnoty získáme zjištěním ozvěny za pomoci generování vysokofrekvenčních zvukových vln. K přeměně energie na ultrazvukové vlny se používá ultrazvukový převodník. Měření vzdálenosti pomocí ultrazvuku se využívá v obtížněji přístupných oblastech. Infračervené senzory měří za pomoci dvou nebo i více infračervených diod, kde jedna dioda slouží jako vysílač a druhá jako přijímač. Vysílač vyšle signál o dané frekvenci a pokud signál narazí na překážku, odrazí se zpět k přijímači. Infračervené senzory dokážou měřit do vzdálenosti několika desítek centimetrů a využívají se především v robotických projektech, zato ultrazvukové senzory mohou měřit vzdálenost i na několik metrů. Existují ovšem i senzory pro měření vzdálenosti, které pracují na odlišném principu.



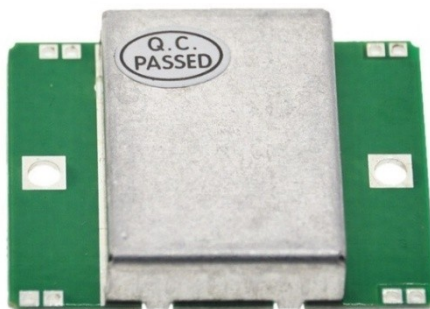
Obrázek 1.1: *Ultrazvukový senzor HC-SR04[1] - upraveno*

1.2 Senzory teploty, vlhkosti a barometrického tlaku

Tyto senzory jsou vhodné jako součást meteostanice, termostatu nebo do průmyslových systémů. Senzory dokážou měřit teplotu obvykle v rozsahu -40 až $+85$ °C s přesností 1 °C až 0,3 °C u těch přesnějších. Rozsah měření relativní vlhkosti se pohybuje v rozsahu 0 až 100 % s přesností měření okolo 2 %. Měřit barometrický tlak je obvykle možné v rozsahu 300 až 1100 hPa. Senzory pro měření barometrického tlaku lze také využít jako výškoměr s poměrně velkou přesností. Senzory se mohou lišit v zobrazení naměřené hodnoty, kde se hodnoty mohou pohybovat s rozlišením na jednotky, desetiny nebo setiny měřené veličiny.[2]

1.3 Senzory pohybu

Senzory pohybu se využívají k detekci pohybu a v případě, že pracují na principu Dopplerova radaru i k měření rychlosti těles. Senzory dokážou detekovat pohyb od vzdálenosti 1 mm až do několika metru. Senzory se mohou lišit velikostí úhlu snímaného okolí. Mikrovlnné moduly nemohou být využívány ve větším počtu ve stejné oblasti, jinak dojde k vzájemnému rušení. Senzory detekce pohybu můžeme využít pro automatické otevírání dveří nebo jako bezpečnostní alarm do domácnosti, chalupy, firmy.



Obrázek 1.2: *Doppler radar HB100[3] - upraveno*

1.4 Senzory hluku

Senzory hluku jsou většinou založeny na zesilovači a citlivém mikrofonu. Výstupem pak bývá analogova, ale i digitální hodnota úrovně zvuku. Deska pro zvukový senzor může obsahovat indikační LED diody, které se rozsvítí při zaznamenání hluku přesahujícího nastavenou hranici. Pro nastavení hranice hluku zde může být malý více otáčkový potenciometr, který zajišťuje přesné nastavení kontrolované úrovně hluku. Tyto senzory lze využít například pro elektronické chůvičky nebo bezpečnostní systémy.

1.5 Senzory světla a barev

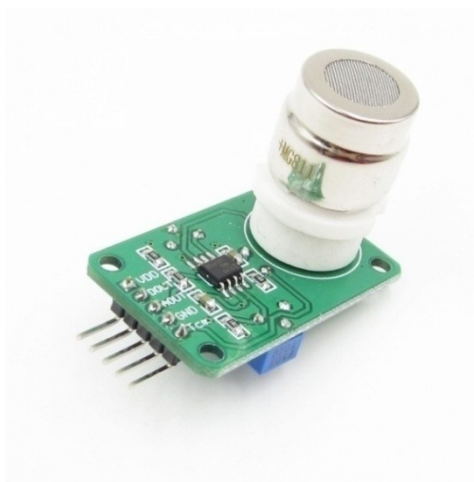
Senzory sloužící pro měření intenzity světla, barev a UV indexu. Měřením UV indexu může například varovat uživatele před úpalem. Díky měření intenzity světla můžeme nastavit správné osvětlení v místnosti. Princip měření barev může u některých senzorů fungovat způsobem, kdy dochází k osvětlení detekovaného předmětu LED diodou a světlo se poté odrazí zpět do senzoru. V senzoru se nacházejí detekční fotodiody a díky tomu můžeme ze senzoru získat informace o barevných složkách RGB. Senzory pro detekci barev je možné využít v různých zařízeních, jako je zařízení pro třídění barevných předmětů nebo ke zjištění barvy okolního osvětlení. Nevýhodou těchto senzorů je, že na přesnost detekce má velký vliv intenzita a barva okolního osvětlení nebo vzdálenost zkoumaného objektu.



Obrázek 1.3: RGB senzor TCS34725[4] - upraveno

1.6 Senzory plynů, kouře a kvality ovzduší

U tohoto typu senzorů je velmi rozhodující cena, kdy za vyšší cenu můžeme pořídit opravdu špičkové měřiče. Pak jsou zde ty levnější, které také najdou své využití, ovšem měření není příliš přesné. Senzory dokážou měřit jeden nebo i více vybraných plynů. Existuje mnoho senzorů, které reagují na oxid uhelnatý nebo na metan, propan, amoniak, oxidy dusíku, benzen, oxid uhličitý a další. Aktivním prvkem těchto senzorů bývá tenká vrstva oxidu cínitého, jejíž odpor se mění s koncentrací zmíněných plynů. Před měřením je nutné nechat senzory zahřát na pracovní teplotu, což může trvat i několik minut. Zároveň je nutné dát pozor na to, kde senzor umístíme, protože pracovní teplota senzoru se pohybuje okolo $+40\text{ }^{\circ}\text{C}$. Senzory můžeme uplatnit v projektech, které se zabývají měřením koncentrace plynů a kouře v ovzduší, jak v domácnosti, tak i na pracovišti.



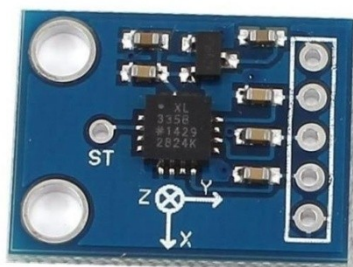
Obrázek 1.4: Detektor oxidu uhličitého MG811[5] - upraveno

1.7 Proudové senzory

Princip měření elektrického proudu může být různý, funguje například na principu měření úbytku napětí na bočníku nebo na principu Hallova jevu, kdy integrovaný obvod umístěný na modulu generuje elektrické napětí na základě okolního elektrického a magnetického pole, které vytváří elektrický proud procházející svorkami. Senzory umožňují měřit proud v obou směrech, tedy kladný i záporný proud. Tyto senzory můžeme využít při realizaci vlastního měřiče proudu nebo přímo multimetru. [23]

1.8 Gyroskopy a akcelerometry

Gyroskopické senzory slouží k měření úhlové rychlosti a pro navigaci, dokážou měřit rychlost otáčení v jedné nebo až třech osách. Akcelerometry jsou elektromechanická zařízení, která dokážou měřit zrychlení opět až ve třech osách. Využívají se pro zjištění polohy, rychlosti a snímání vibrací. Tyto senzory lze aplikovat tam, kde je potřeba znát polohu objektu v prostoru, jako je třeba podomácku vyrobená helikoptéra nebo letadýlko. Využívají se také při orientaci displeje v mobilu.



Obrázek 1.5: Akcelerometr ADXL335[6] - upraveno

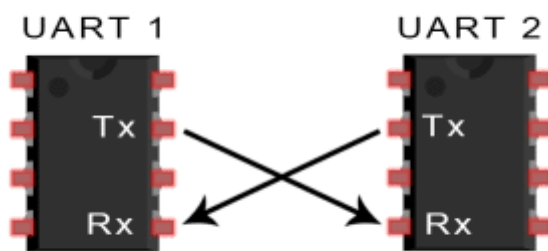
2 Komunikační protokoly

Protokol definuje, jakým způsobem senzor komunikuje s deskou mikrokontroléru. Protokol stanoví způsob zapojení vodičů a jakým způsobem můžeme zažádat o navázání komunikace nebo o měření. I když existuje nepřehledné množství různých senzorů a zařízení, existuje omezený počet používaných protokolů. [7]

Protokoly SPI, I2C a UART jsou ideální pro vzájemnou komunikaci mezi mikrokontroléry a pro komunikaci mezi mikrokontrolérem a senzorem, kde není potřeba přenášet velké množství dat s vysokou rychlostí. Protokoly SPI, I2C a UART jsou pomalejší než protokoly USB, Ethernet, Bluetooth a WiFi, ale jsou mnohem jednodušší a využívají méně hardwarových a systémových prostředků. [8]

2.1 UART

UART je zkratka pro Universal Asynchronous Receiver Transmitter. Hlavním úkolem UART je odesílat a přijímat sériová data. Využívá se například pro připojení GPS modulu nebo Bluetooth modulu. Pro přenos dat se využívají dva vodiče, kde pin Tx slouží pro odesílání dat a pin Rx pro příjem dat. [9]



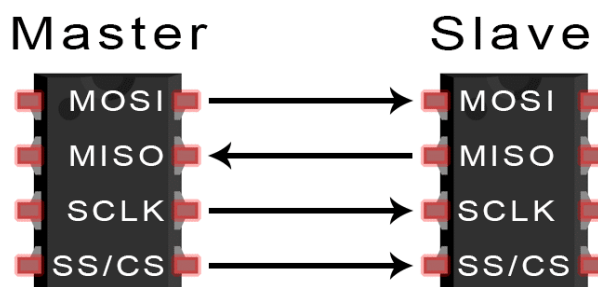
Obrázek 1.6: Základní schéma zapojení UART [9] - upraveno

UART nepoužívá hodinový signál pro synchronizaci výstupních bitů. Místo toho UART přidává start a stop bity k přenášenému datovému paketu. Při obdržení startovacího bitu, začne zařízení číst bity s definovanou přenosovou rychlostí. Obě UART zařízení by měly pracovat na stejné přenosové rychlosti. Datový paket se přenáší sériově a skládá se ze start bitu, pěti až devíti datových bitů, volitelného paritního bitu a stop bitů. Pomocí paritního bitu můžeme zjistit, zda byla data během přenosu změněna. [9]

Výhodou UART je, že využívá pouze dva vodiče a není nutný hodinový signál. Nevýhodou je omezení datového rámce na maximálně 9 bitů a také nepodporuje více Master a Slave zařízení. [9]

2.2 SPI

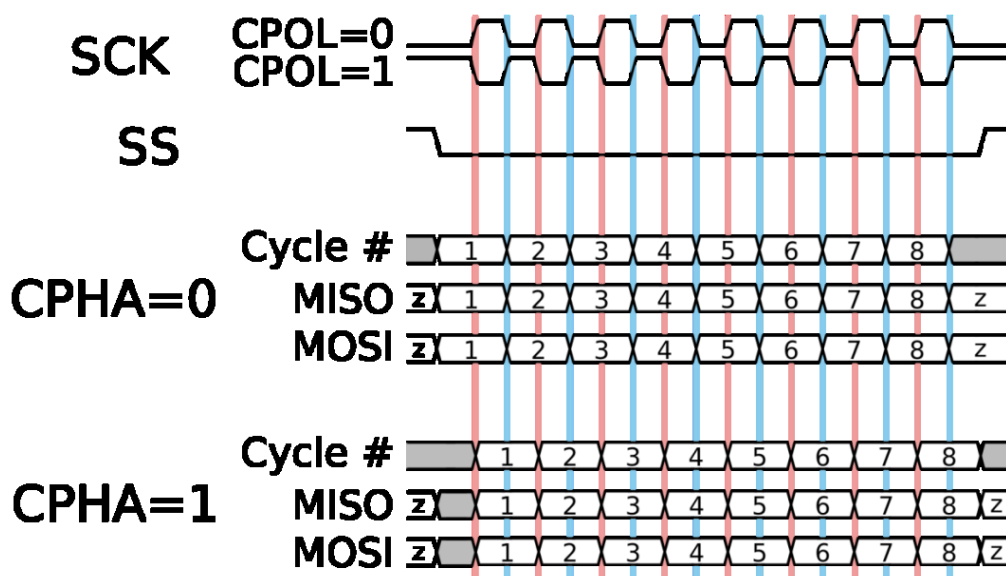
Serial Peripheral Interface je synchronní sériové komunikační rozhraní specifikované pro komunikaci na krátkou vzdálenost a to především do vestavěných systémů. Výhodou SPI je, že data mohou být přenášena bez přerušení. Může být nepřetržitě odeslán nebo přijat libovolný počet bitů. Komunikace probíhá pomocí Master - Slave architektury. Master je řídicí zařízení, obvykle mikrokontrolér, zatímco Slave je senzor, displej nebo paměťový čip. Slave přebírá instrukce od Master. Zařízení Slave může vždy komunikovat pouze se zařízením Master. Jelikož zařízení sdílejí hodinový signál, je tento protokol označován jako synchronní. Na obrázku 1.7 můžeme vidět nejjednodušší konfiguraci SPI, kdy máme jedno Master a Slave zařízení. [8]



Obrázek 1.7: Základní schéma zapojení SPI [8] - upraveno

Na pinu MISO (Master In Slave Out) Slave vysílá a Master přijímá. Na pinu MOSI (Master Out Slave In) vysílá Master a Slave přijímá. SCLK (Serial Clock) je pin pro hodinový signál. SS (Slave Select) slouží pro výběr zařízení, se kterým bude Master komunikovat.

Hodinový signál synchronizuje výstup datových bitů. V každém hodinovém cyklu se přenáší jeden bit dat, takže rychlost přenosu dat je dána frekvencí hodinového signálu. Hodinový signál můžeme změnit pomocí taktovací fáze hodin (CPHA) a polarity hodin (CPOL). Tyto dvě vlastnosti společně definují, kdy jsou bity na výstupu a kdy jsou vzorkovány. CPOL označuje klidový stav, který může být buď v nízkém stavu nebo vysokém stavu. CPHA označuje okraj hodinového signálu, na kterém jsou data zachycena. V závislosti na CPHA jsou data zachycena na náběžné nebo na klesající hraně. Díky těmto různým stavům CPOL a CPHA existují čtyři způsoby přenosu dat. [10]

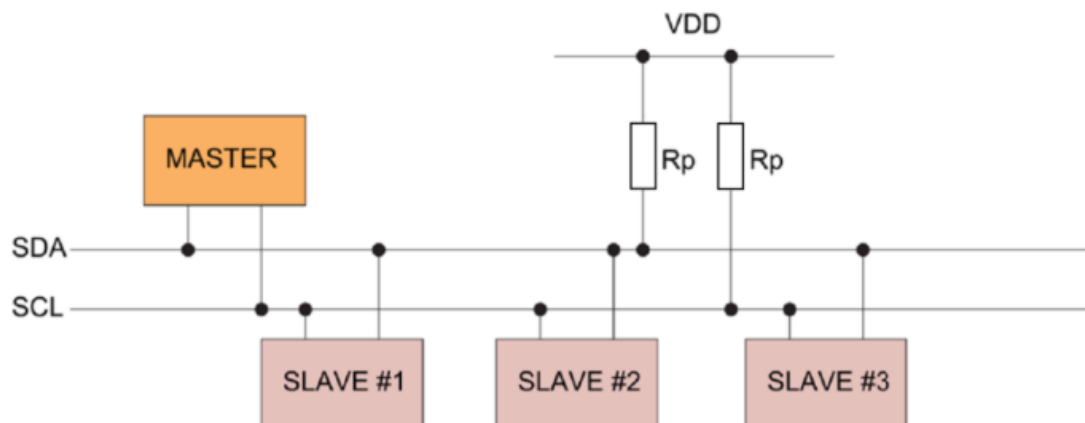


Obrázek 1.8: Časový diagram zobrazující úroveň a posun hodinového signálu [11] - upraveno

Výhodou SPI je, že zde nejsou start a stop bity, takže data mohou být přenášena bez přerušení. Díky odděleným linkám MISO a MOSI mohou být data odesílána a přijímána současně. Nevýhodou je nutnost zapojení čtyř vodičů a také zde není definován způsob pro kontrolu chyb, Master by mohl vysílat data na prázdno a nevěděl by o tom. SPI lze využít pouze pro přenos na krátké vzdálenosti. [8]

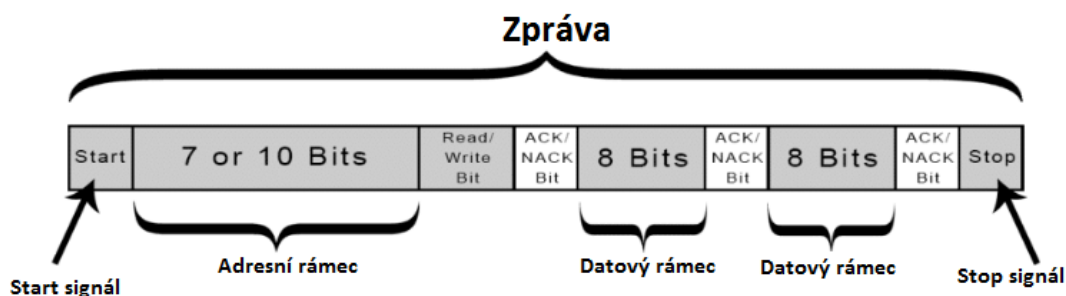
2.3 I2C

I2C byla vyvinuta firmou Philips Semiconductor. I2C využívá pro komunikaci pouze dva vodiče a umožňuje obousměrnou komunikaci. Sběrnice se skládá z linky SDA (Serial Data), která slouží pro obousměrný přenos dat a linky SCL (Serial Clock), která nese hodinový signál a slouží k synchronizaci všech datových přenosů na sběrnici I2C. Na tyto dvě linky jsou pak připojeny všechna I2C zařízení. Dále je zde třetí vodič, který slouží pro napájení připojených zařízení a čtvrtý vodič je zem. Linky SDA a SCL pracují v napěťových logických úrovních 0 a 1. Pro to, abychom vytvořili správně fungující I2C sběrnici, na které jsou připojena všechna I2C zařízení je potřeba propojit linky SCL a SDA s napájecím vodičem pomocí rezistorů, těmto rezistorům se říká pull-up rezistory. Díky těmto dvěma rezistorům je udržován klidový stav v úrovni logické 1.[12] Pro celou sběrnici je zapotřebí pouze jedné dvojice rezistorů. Pokud by rezistory chyběly sběrnice nebude fungovat správně. Běžné hodnoty odporů jsou 1,8 kΩ, 4,7 kΩ nebo 10 kΩ. Pokud ovšem chceme pomocí I2C připojit pouze jeden senzor, není potřeba připojovat žádné rezistory.



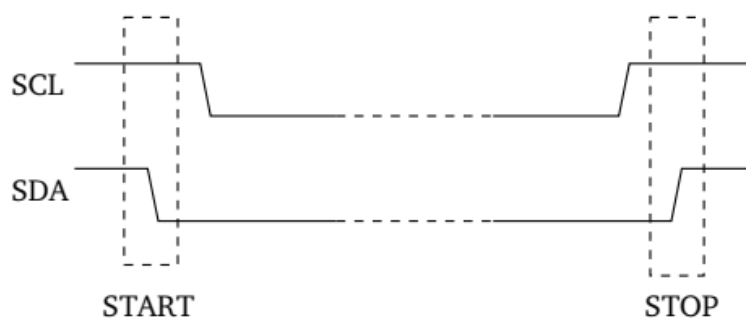
Obrázek 1.9: Zapojení zařízení na I2C sběrnici [13] - upraveno

Na obrázku 1.10 můžeme vidět, že jsou data přenášena ve zprávách. Zprávy jsou rozděleny do rámců. Každá zpráva obsahuje adresní rámec, kde je binární adresa podřizovaného zařízení s nimž chce master komunikovat. Na konci rámce je jeden bit, který určuje, zda chce Master od Slave data číst nebo data zapisovat. Dále následuje jeden nebo více datových rámců, kde jsou přenášena data. Zpráva také obsahuje speciální signály Start a Stop, které slouží pro oddělení zpráv, tyto speciální signály se odlišují od datových bitů. Každým rámcem musí být potvrzen bitem ACK, pokud byl rámec úspěšně přijat, případně bitem NACK. [14]



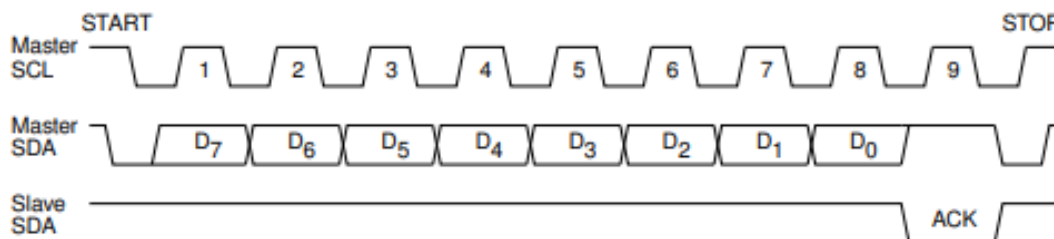
Obrázek 1.10: Přenášená I2C zpráva [14] - upraveno

Na obrázku 1.11 můžeme vidět, že pro Start bit se linka SDA přepíná z úrovně vysokého napětí na úroveň nízkého napětí, před tím než se linka SCL přepne z vysoké úrovně na nízkou. Pro Stop bit se linka SDA přepíná z nízkého napětí na úroveň vysokého napětí poté co se linka SCL přepne z nízké úrovně na vysokou. [14]



Obrázek 1.11: Průběh signálů na I2C [12] - upraveno

Na sběrnici je obvykle jedno zařízení Master a další zařízení, která jsou označována jako Slave. Master řídí příjem a vysílání zpráv. Navázání komunikace zahájí Master tím, že pošle na sběrnici adresu zařízení s nimž chce komunikovat a také bit pro rozlišení zápisu nebo čtení. Zařízení na sběrnici porovnávají tuto adresu, pokud je to jejich adresa potvrdí bitem ACK. Master poté pokračuje v režimu vysílání nebo přijímání dat. Na obrázku 1.12 můžeme vidět průběh přenosu jednoho bajtu, kde se jako první přenáší bit s nejvyšším významem.



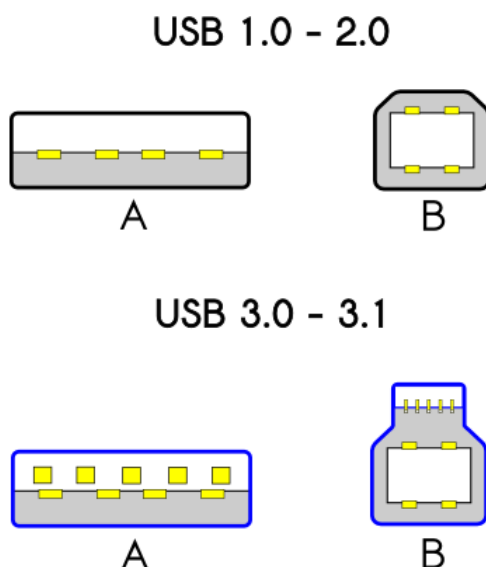
Obrázek 1.12: Přenos jednoho bajtu s potvrzením [12] - upraveno

Protože I2C používá k adresování sedmibitovou adresu je k dispozici 128 adres pro připojená zařízení. Každé zařízení na sběrnici musí mít svou unikátní adresu. Výhodou I2C je, že používá pouze dva vodiče, podporuje více zařízení Master a Slave a na rozdíl od UART a SPI víme, zda přenášená data v pořádku dorazila. Nevýhodou je, že Master může komunikovat v jeden okamžik pouze s jedním Slave a jen jedním směrem. Přenos dat je u I2C pomalejší než u SPI a velikost datového rámce je omezena na 8 bitů. [14]

2.4 USB

USB je univerzální sériová sběrnice a slouží pro připojení zařízení jako je myš, klávesnice, tiskárna, fotoaparát a mnoho dalších. Garantovaná délka pro správný přenos dat je 5 m. Základními vlastnostmi USB je nezávislost na používané platformě, možnost připojit a odpojit zařízení za chodu, garantována přenosová kapacita a zpoždění. Obsahuje nástroje pro detekci chyb a znovu zaslání poškozených paketů. Umožňuje napájet méně energeticky náročná zařízení přímo ze sběrnice a lze připojit až 127 zařízení současně. Při připojení více zařízení je přenosová kapacita sdílena se všemi USB zařízeními. [15]

USB sběrnice má stromovou strukturu a zařízení lze rozdělit na USB hostitele, USB zařízení a USB rozbočovač. USB hostitel je hardware a software, který je součástí počítače. Řídí spolu s operačním systémem veškerou komunikaci s připojenými zařízeními a jejich nastavení. USB zařízení je zařízení, které lze připojit k počítači pomocí USB a USB rozbočovač je speciální zařízení, které vytváří nové přípojné body. Ve stromové struktuře může být pouze jeden USB hostitel. USB zařízení spolu nemohou komunikovat přímo, ale pouze přes USB hostitele. [15]

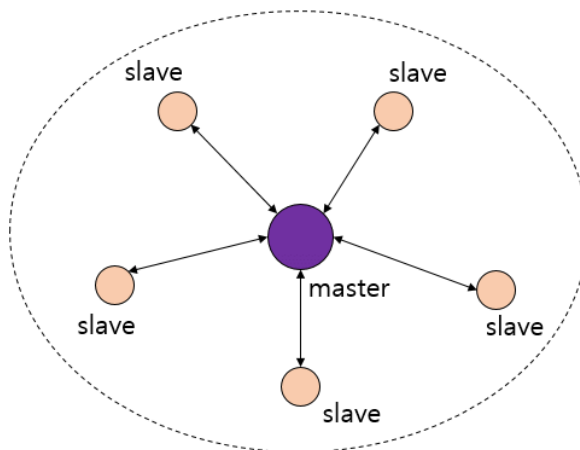


Obrázek 1.13: Základní typy USB konektorů [16] - upraveno

USB vzniklo v několika verzích. USB ve verzi 1.1 dosahuje rychlosti 12 Mbit/s. Následovala verze 2.0, která dosahuje rychlosti 480 Mbit/s. USB sběrnice ve verzích 1.0 - 2.0 využívá celkem 4 vodiče. Červený vodič pro napájecí napětí 5 V, černý pro zem a dvojici datových vodičů, kde jeden je zelený druhý bílý. Ve verzi USB 3.0 - 3.1 je použito 9 vodičů a přenosová rychlost je 5 Gbit/s. [16]

2.5 Bluetooth

Bluetooth je standardizovaný protokol, který pracuje na frekvenci 2,4 GHz v nelicencovaném frekvenčním pásmu ISM. Je ideální pro bezdrátové přenosy mezi elektronickými zařízeními s krátkým dosahem a nízkou spotřebou. Bluetooth je vhodný jako bezdrátová náhrada sériových komunikačních rozhraní. Sítě Bluetooth jsou označovány jako Piconet a využívají model Master - Slave, kde je jedno Master zařízení a až 7 Slave zařízení. Každé Slave zařízení může být v Piconetu připojeno pouze k jednomu Master zařízení. Master řídí komunikaci v celém Piconetu. [17]



Obrázek 1.14: Bluetooth topologie Piconet [18] - upraveno

Vysílací výkon Bluetooth modulu je definován výkonovou třídou. V tabulce 1.1 můžeme vidět, rozdělení Bluetooth do tříd.

Tabulka 1.1: Rozdělení Bluetooth do tříd dle výkonu

Číslo třídy	Max. povolený výkon (dBm)	Max. povolený výkon (mW)	Přibližný dosah
1	20	100	100 m
2	4	2,5	10 m
3	0	1	<1 m

Každé Bluetooth zařízení má jedinečnou adresu o velikosti 48 bitů, obvykle ve tvaru dvanácti místné hexadecimální hodnoty, kde významnější polovina bitů adresy je jedinečný identifikátor výrobce. Méně významnějších 24 bitů určuje jedinečnost zařízení. Místo identifikace podle této adresy mívají zařízení uživatelský přívětivá jména. Pro to, aby spolu mohla dvě zařízení komunikovat musí dojít ke spárování. Když jsou dvě Bluetooth zařízení vzájemně spárována, mohou vždy automaticky navázat spojení, když jsou u sebe dostatečně blízko. Zařízení sdílejí své adresy, profily, názvy a společný tajný klíč. Pro spárování zařízení je obvykle potřeba ověřovací proces, který například u verze 2.0 zahrnuje zadání čtyřmístného číselného PIN kódu. [17]

2.5.1 Bluetooth profily

Bluetooth může mít různé profily, které definují, jaké data Bluetooth modul vysílá. Pro to, aby byla dvě zařízení kompatibilní, musí podporovat stejné profily.

Profil sériového portu je vhodný pro posílání dat mezi dvěma zařízeními, pokud chceme pomoci Bluetooth technologie nahradit sériové komunikační rozhraní RS-232 nebo UART. Pomoci tohoto profilu mohou připojená zařízení komunikovat stejně, jako by byla propojena pomocí linky RX a TX.

Human Interface Device je profil pro zařízení jako jsou myši, joysticky, klávesnice nebo moderní herní ovladače. Cílem tohoto profilu je nahradit kabely RS-232 a USB kabely.

Handsfree profil se využívá v autech se zabudovaným audio systémem. Umožňuje funkce jako je přijímání, odmítání a zavěšení hovorů bez nutnosti mít mobil v ruce.

Advanced Audion Distribution Profile je profil pro jednosměrný přenos zvuku, tedy pouze z jednoho zařízení do druhého, kvalita zvuku je však mnohem vyšší. Tento profil je vhodný pro bezdrátové audio přenosy. [17]

Od svého vzniku se Bluetooth neustále vyvíjí a vznikají stále nové verze. Tato technologie se postupně rozšířila a nyní ji využívají nejrůznější typy zařízení. Bluetooth není určitě zcela ideální pro bezdrátovou komunikaci, ale vyniká při práci v krátkých vzdálenostech a také snadným procesem připojení.

3 Návrh řešení automatizovaného sběru dat pomocí senzorů

3.1 Návrh řešení

V této kapitole se budu zabývat návrhem řešení automatizovaného sběru dat ze senzorů, připojených k mikrokontroléru Arduino a následným přenosem dat do počítače s OS Linux. Mé řešení se skládá z vývojové desky Arduino Uno, senzoru pro měření teploty, vlhkosti a barometrického tlaku BME280, senzoru pro měření intenzity světla BH1750, senzoru napětí a proudu INA3221 a Bluetooth modulu HC-05 pro bezdrátový přenos dat. Jelikož na notebooku, kde jsem pracoval na této bakalářské práci používám operační systém Windows, vytvořil jsem si pomocí programu Oracle VM VirtualBox virtuální počítač, na který jsem nainstaloval OS Ubuntu 18.04.2 LTS. Díky tomuto kroku se data přenášejí pomocí Bluetooth modulu nejprve na Bluetooth sériový port fyzického počítače s OS Windows a poté jsou data přeposílána z tohoto sériového portu na sériový port virtuálního počítače.



Obrázek 1.15: Zjednodušený návrh řešení

K naprogramování mikrokontroléru Arduino bylo použito vývojové prostředí Arduino IDE. Po nahrání požadovaného kódu do Arduina a spárování Bluetooth modulu s počítačem začne Arduino posílat data pomocí Bluetooth modulu na sériový port počítače. Pomocí skriptu napsaného v jazyce Python dochází ke čtení a zpracování dat ze sériového portu. Po úpravě dat do požadovaného tvaru jsou tyto data vložena do databáze pro správu a ukládání časových řad InfluxDB. Pro grafickou vizualizaci naměřených dat propojíme vytvořenou databázi s vizualizačním softwarem Grafana, kde můžeme vidět grafický průběh naměřených dat.

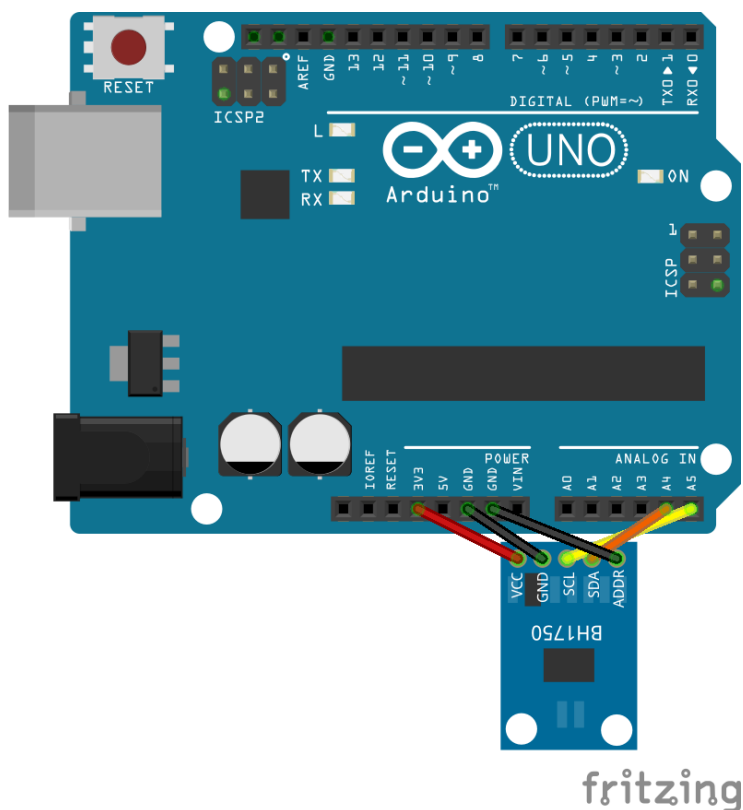
3.2 Kontrola senzoru intenzity světla BH1750

Senzor intenzity světla BH1750 umožňuje měřit intenzitu světla ve svém okolí. Senzor komunikuje přes rozhraní I2C s defaultní I2C adresou 0x23.



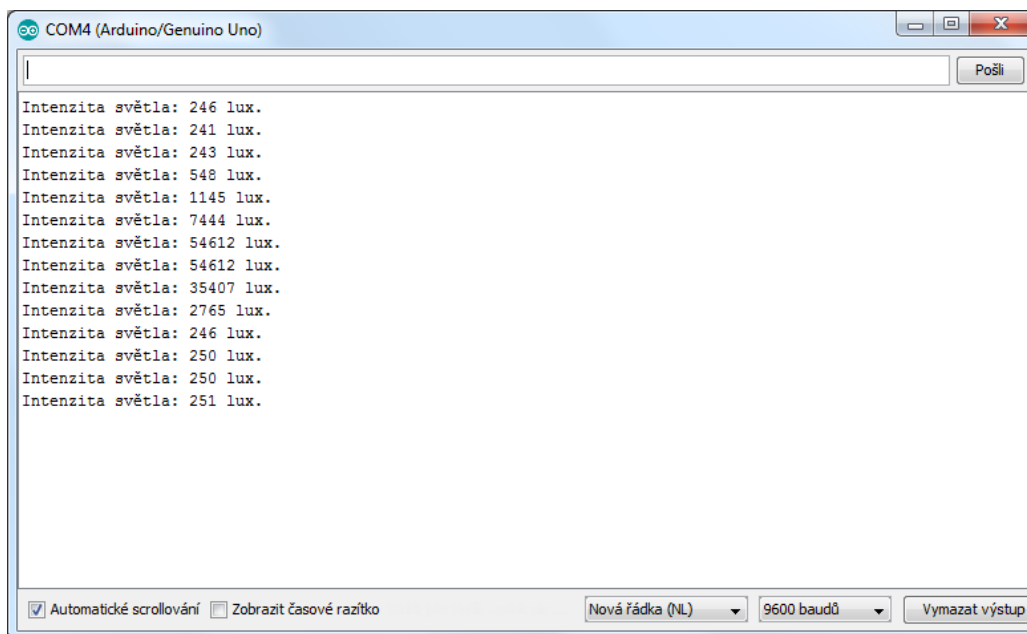
Obrázek 1.16: *Senzor intenzity světla BH1750 [19] - upraveno*

Na obrázku 1.17 můžeme vidět zapojení jednotlivých vodičů, tento obrázek byl vytvořen v softwaru Fritzing. Je potřeba přivést napětí 3,3 V z Arduina na pin VCC, dále propojit piny GND a ADDR se zemí Arduina a piny SCL a SDA s I2C piny Arduina, které jsou u desky Arduino Uno na analogových pinech A5 a A4.



Obrázek 1.17: *Zapojení senzoru BH1750*

Pro usnadnění komunikace pomocí I2C jsem importoval knihovny (Příloha A). Knihovny za nás řeší například zahájení a ukončení komunikace se senzorem a přenesení bajtů, kde jsou uloženy naměřené hodnoty. Knihovna také usnadňuje následný přepočet přijatých hodnot na lx. Na obrázku 1.18 můžeme vidět výpis naměřených hodnot v programu Arduino IDE. Senzor dokáže měřit hodnoty v rozsahu od 1 až 65535 lx. Hodnoty okolo 250 lux byly naměřeny v místnosti při denním světle a vysokých hodnot 54612 lux jsem dosáhl při posvícení světlem od mobilního telefonu na senzor. Kód programu, který jsem použil je v Příloze B.



Obrázek 1.18: Výpis dat ze senzoru BH1750 v programu Arduino IDE.

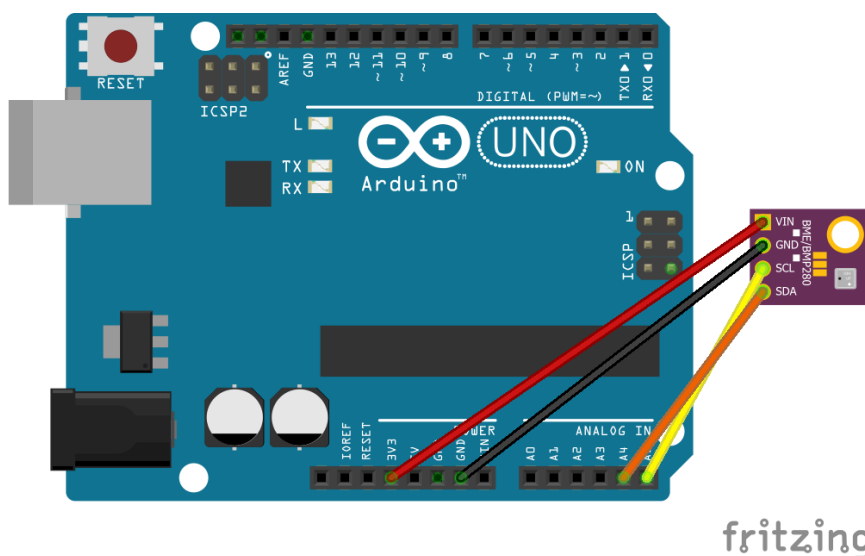
3.3 Kontrola senzoru BME280

Senzor BME280 umožňuje měřit teplotu, vlhkost a barometrický tlak. Podporuje protokoly I2C a SPI. Senzor měří teplotu v rozsahu -40 až +85 °C s přesností jednoho °C. Vlhkost měří v rozsahu 0 až 100 % relativní vlhkosti a tlak v rozsahu 300 až 1100 hPa. Všechny tyto měřené veličiny dokáže senzor vypisovat s rozlišením na setiny. [2]



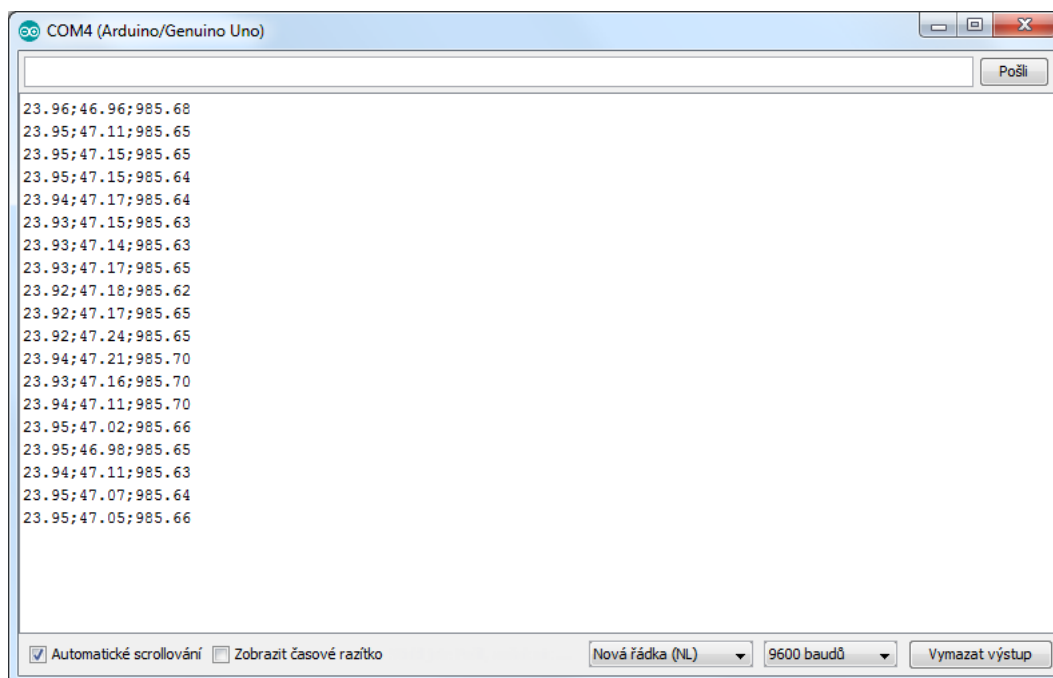
Obrázek 1.19: Senzor BME280 [20] - upraveno

Na obrázku 1.20 můžeme vidět, že je potřeba propojit čtyři piny. Přivedeme napětí 3,3 V z Arduina na pin VCC, dále je potřeba propojit pin GND se zemí Arduina a piny SCL a SDA s I2C piny Arduina, které jsou na pinech A5 a A4.



Obrázek 1.20: Zapojení senzoru BME280

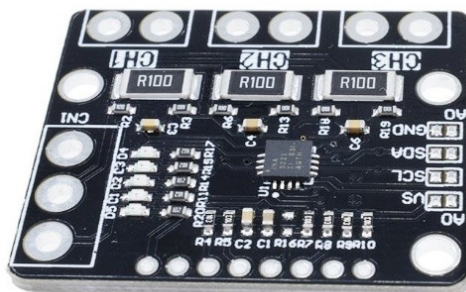
Při psaní kódu v programu Arduino IDE jsem opět využil dostupných knihoven (Příloha C). Na obrázku 1.21 můžeme vidět výpis naměřených hodnot v programu Arduino IDE. Naměřené hodnoty jsou od sebe odděleny interpunkčním znaménkem středník. Jako první hodnota zleva je teplota ve °C, následuje relativní vlhkost v % a barometrický tlak hPa. Kód programu viz Příloha D.



Obrázek 1.21: Výpis dat ze senzoru BME280 v programu Arduino IDE

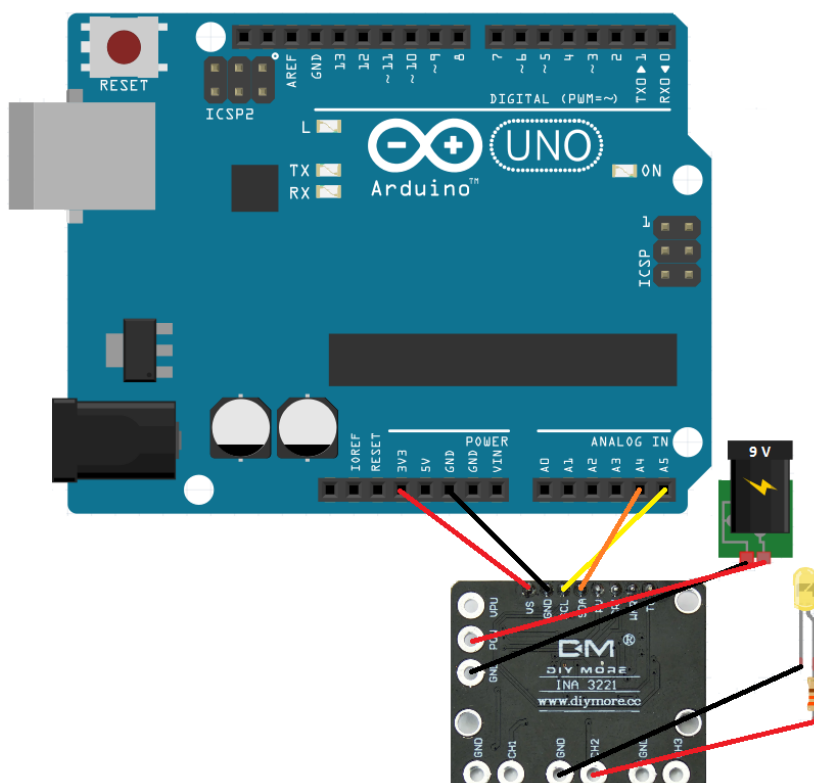
3.4 Kontrola senzoru INA3221

Senzor napětí a proudu INA3221 dokáže měřit napětí a proud na třech spotřebičích díky třem měřicím kanálům. K tomuto senzoru je pro měření potřeba připojit externí napájecí zdroj pro všechny tři kanály. Senzor komunikuje přes rozhraní I2C.



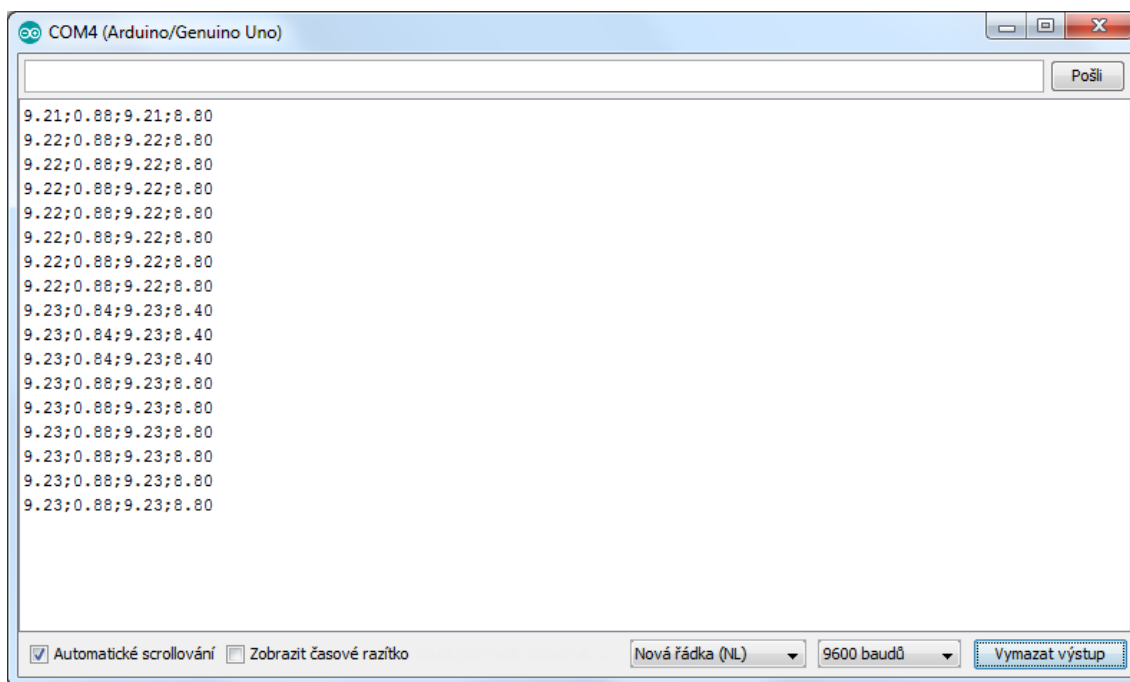
Obrázek 1.22: Senzor INA3221 [21] - upraveno

Na obrázku 1.23 můžeme vidět, že je potřeba propojit čtyři piny. Přivedeme napětí 3,3 V z Arduino na pin VCC, dále propojíme pin GND se zemí Arduino a piny SCL a SDA s I2C piny Arduino, kde pin SCL je pod analogovým pinem A5 a pin SDA pod analogovým pinem A4. Na druhém kanále je pro měření připojena LED dioda.



Obrázek 1.23: Zapojení senzoru INA3221

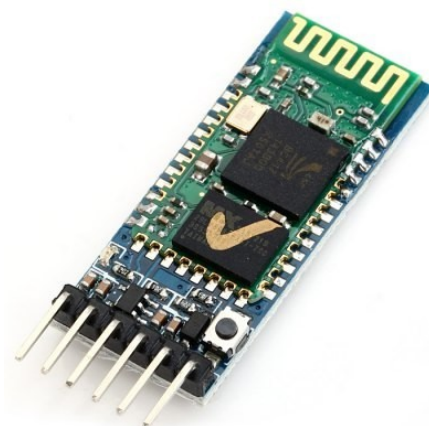
Při psaní kódu v programu Arduino IDE jsem opět pro usnadnění komunikace se senzorem využil dostupné knihovny (Příloha E). Pro měření jsem na druhý kanál připojil LED diodu a jako externí zdroj jsem použil 9 V baterii. Na obrázku 1.24 můžeme vidět výpis naměřených hodnot na druhém kanále. Jako první zleva je hodnota napětí na zdroji ve V, následuje hodnota napětí na bočníku v mV, hodnota napětí na zátěži ve V a hodnota proudu na zátěži v mA. Kód programu je uveden v Příloha F.



Obrázek 1.24: Výpis dat ze senzoru INA3221 v programu Arduino IDE

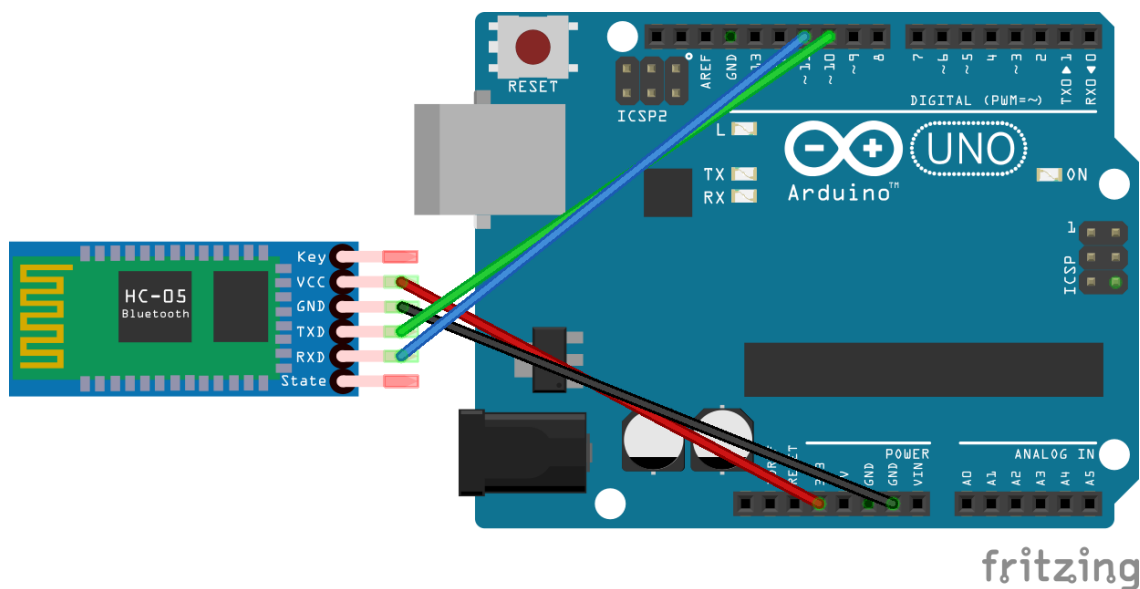
3.5 Kontrola Bluetooth modulu HC-05

Bluetooth modul HC-05 můžeme využít pro odesílání nebo příjem dat, tedy jako vysílač i přijímač. Umožňuje propojit Arduino s počítačem nebo mobilním telefonem. Ve svém návrhu automatizovaného sběru dat budu využívat Bluetooth modul pro odeslání dat z Arduina na sériový port počítače.



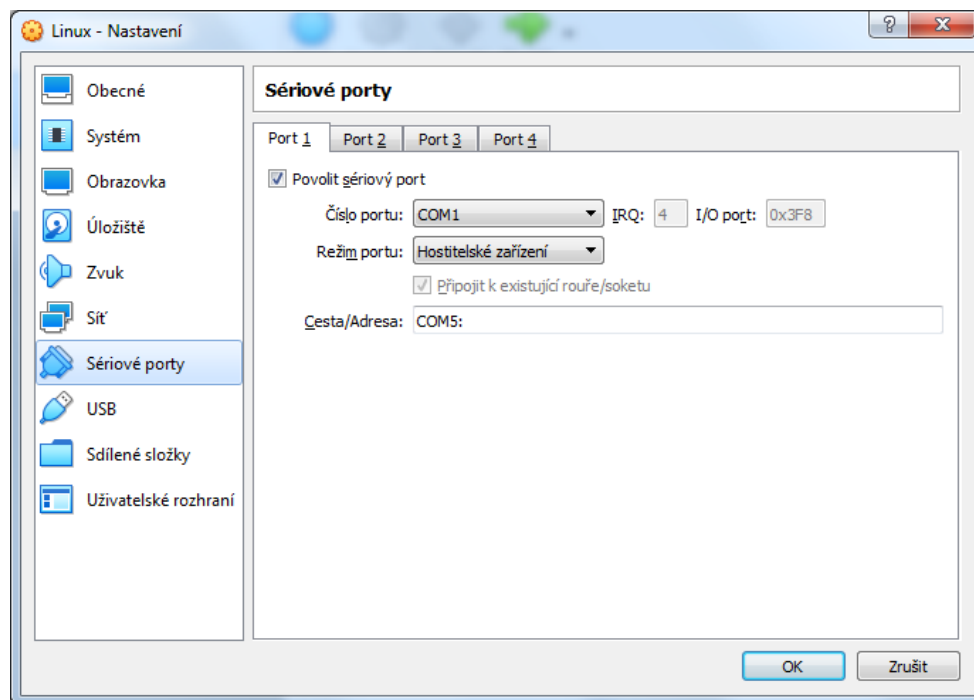
Obrázek 1.25: Bluetooth modul HC-05[22] - upraveno

Z obrázku 1.26 lze vyčíst, že je potřeba přivést napětí 3,3 V z Arduina na pin VCC. Dále je potřeba propojit pin GND se zemí Arduina a datové piny TXD a RXD s digitálními piny D10 a D11, případně s jinými digitálními piny. Tuhle změnu bychom však museli provést i ve zdrojovém kódu, který budeme nahrávat do Arduina.



Obrázek 1.26: Zapojení Bluetooth modulu HC-05

Po nahrání zdrojového kódu do Arduina můžeme odpojit USB kabel, pomocí kterého jsme komunikovali s Arduinem a také ho napájeli. Jako nový zdroj napájení jsem zvolil 9 V baterii, kterou jsem připojil k Arduinu. Nyní je potřeba spárovat Bluetooth modul s počítačem, po úspěšném spárování začneme pomocí Bluetooth modulu přijímat data na sériový port COM5 fyzického počítače. My však potřebujeme odesílat data do našeho virtuálního počítače, to provedeme tak, že v nastavení programu Oracle VM VirtualBox zatrhneme možnost povolit sériový port a na port COM1 přiřadíme adresu portu COM5. Toto nastavení můžeme vidět na obrázku 1.27.



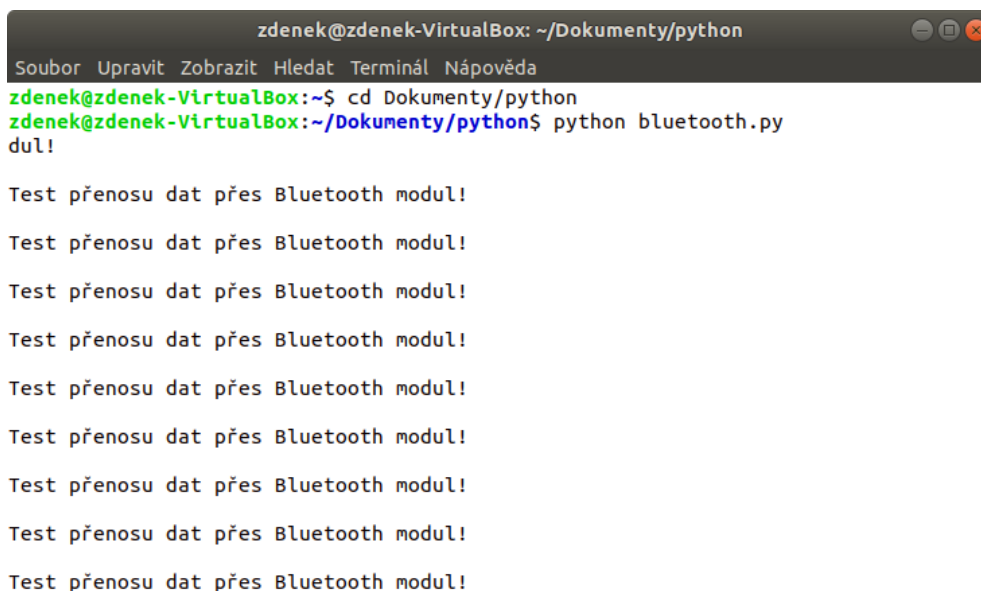
Obrázek 1.27: Nastavení sériového portu v Oracle VM VirtualBox

Nyní si ve virtuální počítači pomocí jednoduchého skriptu v Pythonu, který můžeme vidět na obrázku 1.28 ověříme, zda přicházejí data na sériový port `/dev/ttyS0` ve virtuálním počítači. Můžeme vidět, že v nekonečné smyčce čteme data ze sériového portu a ukládáme je do proměnné `nacteny_radek`. Proměnnou poté vypisujeme.



Obrázek 1.28: Skript pro kontrolu dat na sériovém portu

Po spuštění skriptu v terminálu můžeme na obrázku 1.29 vidět, že posílaná testovací data přicházejí na sériový port virtuálního počítače. Tímto jsme si ověřili, že posílání dat přes Bluetooth modul do virtuálního počítače funguje tak, jak potřebujeme. Kód nahraný do Arduina pro toto testování viz Příloha G.



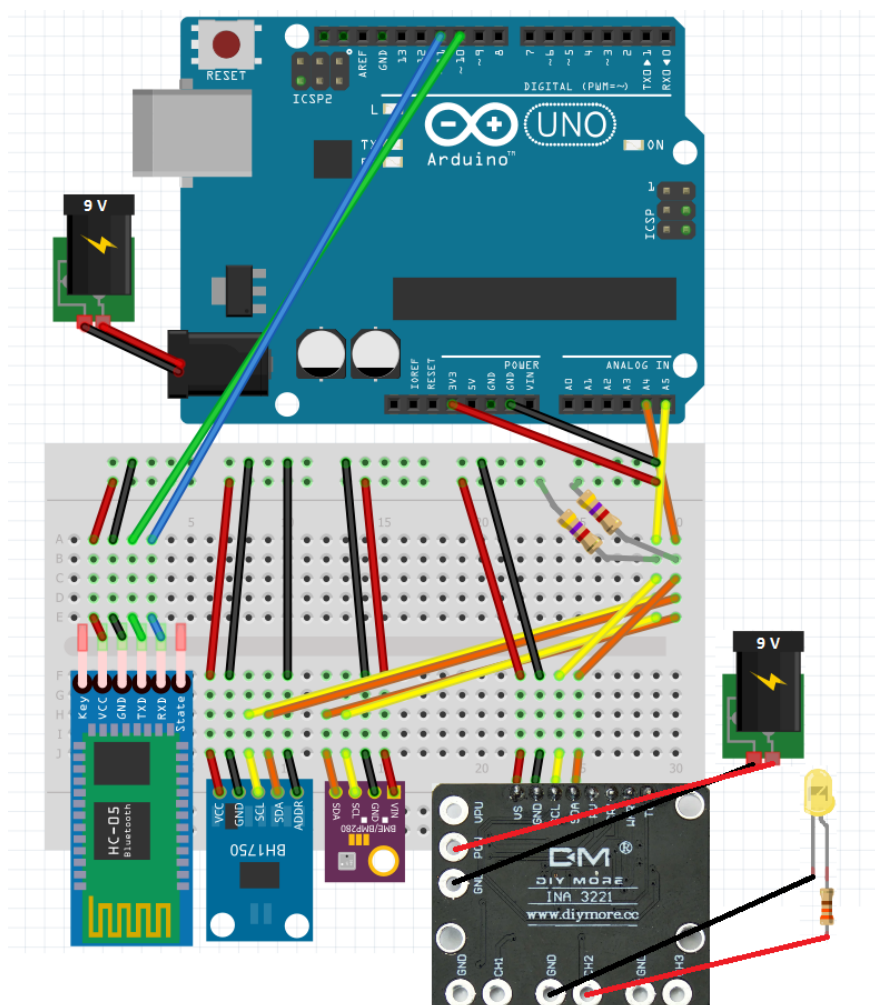
Obrázek 1.29: Výpis dat do terminálu pomocí skriptu v Pythonu

3.6 Propojení všech zařízení

Nyní když máme otestovány jednotlivé senzory a Bluetooth modul, můžeme všechna tato jednotlivá zařízení propojit společně s deskou Arduino Uno. Propojení jednotlivých pinů u všech senzorů a Bluetooth modulu je totožné, jako bylo popsáno v předchozích kapitolách.

Vzhledem k tomu, že všechny senzory komunikují přes rozhraní I2C a protokol I2C umožňuje připojení více senzoru, byla za tímto účelem vytvořena I2C sběrnice, kde jsou všechny tyto senzory připojeny.

Jak můžeme vidět na obrázku 1.30, tak jsou všechny senzory propojeny se společným vodičem SDA, který slouží pro přenos dat a společným vodičem SCL pro hodinový signál. Všechny tři senzory i Bluetooth modul jsou propojeny s napájecím napětím 3,3 V Arduino a také se zemí Arduino. Pro správnou funkci I2C sběrnice bylo potřeba propojit vodiče SDA a SCL přes pull-up rezistory o odporu 4,7 k Ω s napájecím napětím 3,3 V Arduino. Ve svém návrhu využívám dvě baterie, kde jedna slouží jako externí zdroj pro senzor INA3221 a druhá pro napájení samotného Arduino.

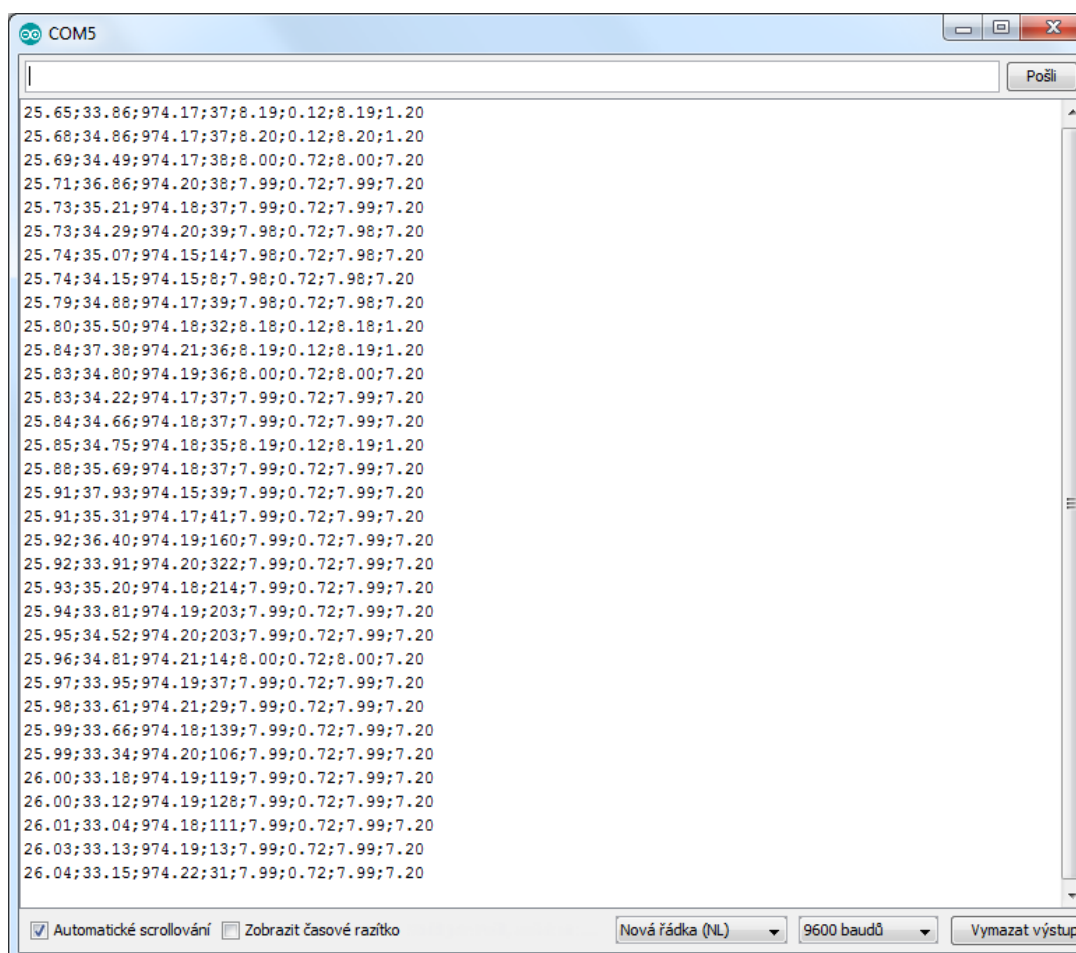


Obrázek 1.30: *Propojení všech zařízení*

K vytvoření kódu pro Arduino byly použity kódy z předchozích kapitol a upraveny tak, aby fungovaly v jednom programu (Příloha H). Výstupem programu jsou naměřené hodnoty na jednotlivých senzorech. Hodnoty jsou vypisovány po řádcích jako textový řetězec, kde každý řádek na výpisu reprezentuje jedno měření. Jednotlivé naměřené hodnoty na senzorech jsou odděleny znaménkem středník. V tabulce 1.2 je vysvětlen význam jednotlivých hodnot zleva na obrázku 1.31. Na obrázku 1.31 je například čtvrtá hodnota zleva intenzita světla v lx.

Tabulka 1.2: Význam vypisovaných hodnot na obrázku 1.31

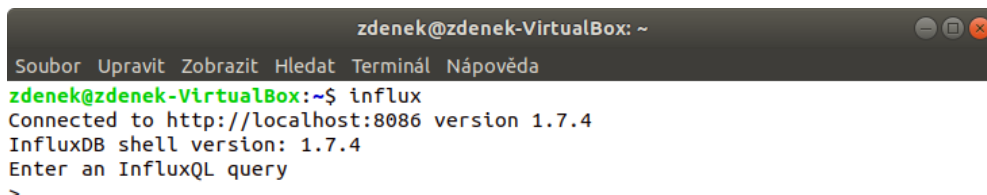
Hodnota	Měřená veličina	Jednotky
1	Teplota	°C
2	Relativní vlhkost	%
3	Barometrický tlak	hPa
4	Intenzita Světla	lx
5	Napětí na zdroji	V
6	Napětí na bočníku	mV
7	Napětí na zátěži	V
8	Proud na zátěži	mA



Obrázek 1.31: Výpis dat ze všech senzorů v programu Arduino IDE

3.7 Vkládání naměřených dat do databáze

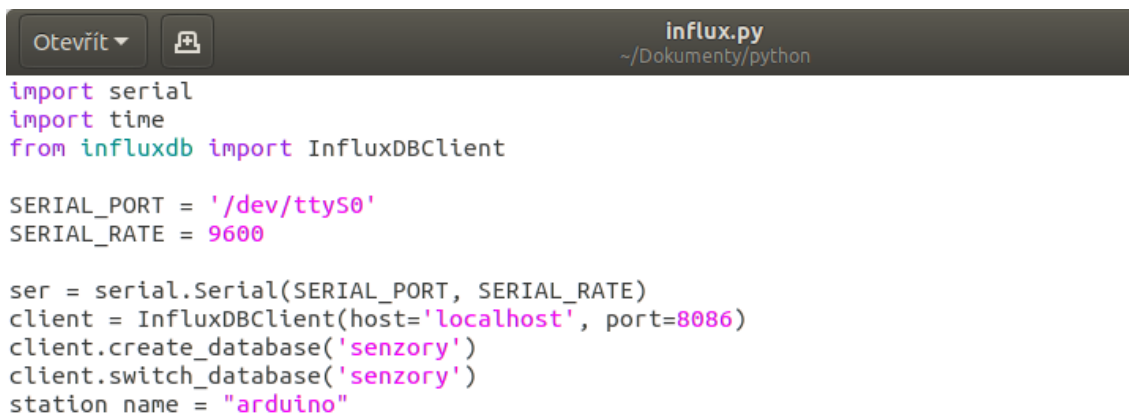
Nyní, když posíláme naměřená data do virtuální počítače je potřeba tyto data někde ukládat. Jako vhodná možnost pro ukládání naměřených dat se ukázala databáze časových řad InfluxDB. Abychom mohli tuhle databázi využívat bylo jí potřeba nejprve nainstalovat na virtuální počítač. Po úspěšné instalaci a spuštění můžeme na obrázku 1.32 vidět, že do databáze můžeme přistupovat na adrese localhost a portu 8086.



```
zdenek@zdenek-VirtualBox: ~  
Soubor Upravit Zobrazit Hledat Terminál Nápověda  
zdenek@zdenek-VirtualBox:~$ influx  
Connected to http://localhost:8086 version 1.7.4  
InfluxDB shell version: 1.7.4  
Enter an InfluxQL query  
>
```

Obrázek 1.32: *Spuštění databáze InfluxDB*

Data jsou do databáze vkládána pomocí skriptu v jazyce Python. Na obrázku 1.33 můžeme vidět část tohoto skriptu, ve kterém vytvoříme novou instanci InfluxDBClient a nastavíme informace o serveru, ke kterému chceme přistupovat. Podle obrázku 1.32 víme, že je to localhost a port 8086. Poté vytvoříme databázi s názvem senzory, do které budeme vkládat naše naměřená data.



```
influx.py  
~/Dokumenty/python  
Otevřít ▾  
import serial  
import time  
from influxdb import InfluxDBClient  
  
SERIAL_PORT = '/dev/ttyS0'  
SERIAL_RATE = 9600  
  
ser = serial.Serial(SERIAL_PORT, SERIAL_RATE)  
client = InfluxDBClient(host='localhost', port=8086)  
client.create_database('senzory')  
client.switch_database('senzory')  
station_name = "arduino"
```

Obrázek 1.33: *Část skriptu pro ukládání dat do databáze*

Dále ve skriptu vytvoříme nekonečnou smyčku, která čte data ze sériového portu a vkládá data do databáze. Jelikož data přicházejí na sériový port jako jeden textový řetězec, který reprezentuje jedno měření, je potřeba řádek rozdělit pomocí interpunkční znaménka středník na jednotlivé naměřené hodnoty. Tyto hodnoty je dále potřeba přetypovat na datový typ float. Hodnoty jsou nyní reprezentovány jako číselný údaj a lze s nimi později pracovat v databázi InfluxDB a vizualizačním softwaru Grafana.

Data vkládáme do databáze jako proměnnou `json_data` ve formátu JSON, kde můžeme nastavit například název měření, název stanice a vytvořit jednotlivé sloupce v databázi, do kterých ukládáme jednotlivé naměřené veličiny. Na konci skriptu proměnnou `json_data` vložíme do databáze a takto pokračujeme v nekonečném cyklu. Databáze si sama k tomuto vloženému údaji přidá časový údaj o přesném času vložení. Celé skript, který byl použit pro ukládání naměřených dat do databáze viz Příloha I.



```

influx.py
~/Dokumenty/python
Uložit

while True:
    nacteny_retezec = ser.readline().decode('utf-8')
    data = nacteny_retezec.split(";")
    teplota = data[0]
    vlhkost = data[1]
    tlak = data[2]
    svetlo = data[3]
    napeti_zdroj = data[4]
    napeti_bocnik = data[5]
    napeti_zatez = data[6]
    proud_zatez = data[7]
    teplota_cislo = float(teplota)
    vlhkost_cislo = float(vlhkost)
    tlak_cislo = float(tlak)
    svetlo_cislo = float(svetlo)
    napetizdroj_cislo = float(napeti_zdroj)
    napetibocnik_cislo = float(napeti_bocnik)
    napetizatez_cislo = float(napeti_zatez)
    proudzatez_cislo = float(proud_zatez)

    json_data = [
        {
            "measurement": "bluetooth",
            "tags": {
                "Station": station_name
            },
            "fields": {
                "Teplota": teplota_cislo,
                "Vlhkost": vlhkost_cislo,
                "Tlak": tlak_cislo,
                "Svetlo": svetlo_cislo,
                "Napeti_zdroj": napetizdroj_cislo,
                "Napeti_bocnik": napetibocnik_cislo,
                "Napeti_zatez": napetizatez_cislo,
                "Proud_zatez": proudzatez_cislo
            }
        }
    ]

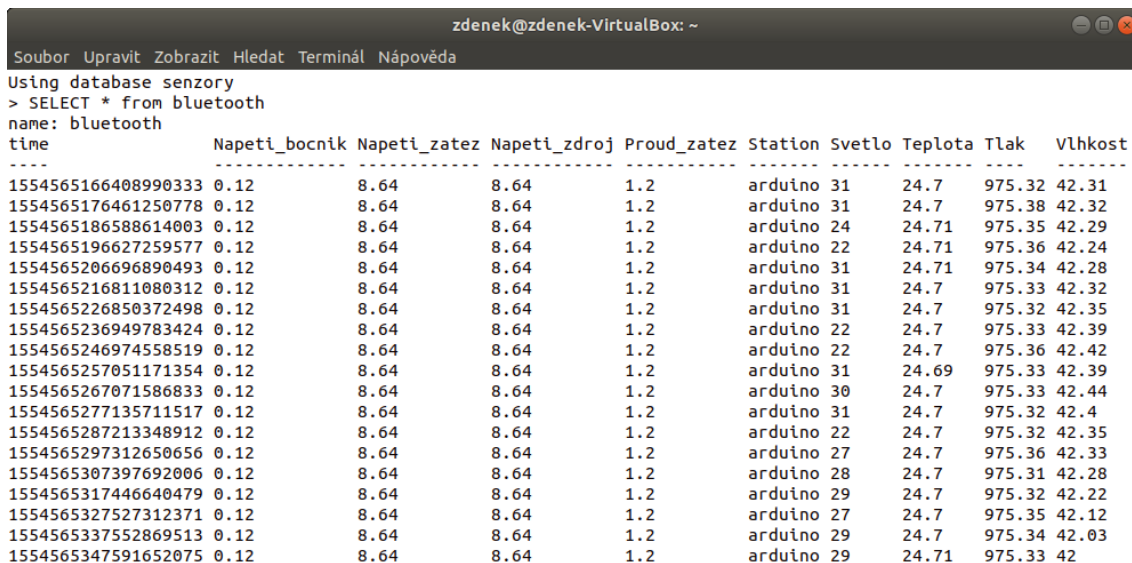
    bResult = client.write_points(json_data)
    print(bResult)

if __name__ == "__main__":
    main()
Python Šířka tabulátoru: 8 ř 56, sl 23 VKL
    
```

Obrázek 1.34: Druhá část skriptu pro ukládání dat do databáze

Po spuštění toho skriptu v terminálu virtuálního počítače dojde k vytvoření databáze s názvem senzory a k pravidelnému vkládání nových naměřených dat do databáze v nekonečném cyklu.

Na obrázku 1.35 můžeme vidět výpis dat z naší vytvořené databáze. V prvním sloupci zleva můžeme vidět časový údaj neboli časové razítko, které si zde databáze InfluxDB přidala ke každému vloženému záznamu. Časová razítka jsou vždy v čase Unixu. Dále vidíme sloupce s našimi naměřenými hodnotami a také sloupec Station, tedy název stanice, který jsme nastavili jako arduino a ten zůstává neměnný.



time	Napeti_bocnik	Napeti_zatez	Napeti_zdroj	Proud_zatez	Station	Svetlo	Teplota	Tlak	Vlhkost
1554565166408990333 0.12	8.64	8.64	8.64	1.2	arduino 31	31	24.7	975.32	42.31
1554565176461250778 0.12	8.64	8.64	8.64	1.2	arduino 31	31	24.7	975.38	42.32
1554565186588614003 0.12	8.64	8.64	8.64	1.2	arduino 24	24	24.71	975.35	42.29
1554565196627259577 0.12	8.64	8.64	8.64	1.2	arduino 22	22	24.71	975.36	42.24
1554565206696890493 0.12	8.64	8.64	8.64	1.2	arduino 31	31	24.71	975.34	42.28
1554565216811080312 0.12	8.64	8.64	8.64	1.2	arduino 31	31	24.7	975.33	42.32
1554565226850372498 0.12	8.64	8.64	8.64	1.2	arduino 31	31	24.7	975.32	42.35
1554565236949783424 0.12	8.64	8.64	8.64	1.2	arduino 22	22	24.7	975.33	42.39
1554565246974558519 0.12	8.64	8.64	8.64	1.2	arduino 22	22	24.7	975.36	42.42
1554565257051171354 0.12	8.64	8.64	8.64	1.2	arduino 31	31	24.69	975.33	42.39
1554565267071586833 0.12	8.64	8.64	8.64	1.2	arduino 30	30	24.7	975.33	42.44
1554565277135711517 0.12	8.64	8.64	8.64	1.2	arduino 31	31	24.7	975.32	42.4
1554565287213348912 0.12	8.64	8.64	8.64	1.2	arduino 22	22	24.7	975.32	42.35
1554565297312650656 0.12	8.64	8.64	8.64	1.2	arduino 27	27	24.7	975.36	42.33
1554565307397692006 0.12	8.64	8.64	8.64	1.2	arduino 28	28	24.7	975.31	42.28
1554565317446640479 0.12	8.64	8.64	8.64	1.2	arduino 29	29	24.7	975.32	42.22
1554565327527312371 0.12	8.64	8.64	8.64	1.2	arduino 27	27	24.7	975.35	42.12
1554565337552869513 0.12	8.64	8.64	8.64	1.2	arduino 29	29	24.7	975.34	42.03
1554565347591652075 0.12	8.64	8.64	8.64	1.2	arduino 29	29	24.71	975.33	42

Obrázek 1.35: Výpis dat z databáze InfluxDB

4 Vizualizace naměřených dat

Aby byla naměřena data více uživatelsky přívětivá a ne jen hodnoty v databázi, rozhodl jsem se využít vizualizační software Grafana. Grafana umožňuje vytvářet typy grafů, které potřebujeme a poskytuje dobré možnosti pro vizualizaci našich dat. Podporuje různé typy databází a zejména s databází InfluxDB, kde ukládáme naměřené hodnoty, spolupracuje velmi dobře.

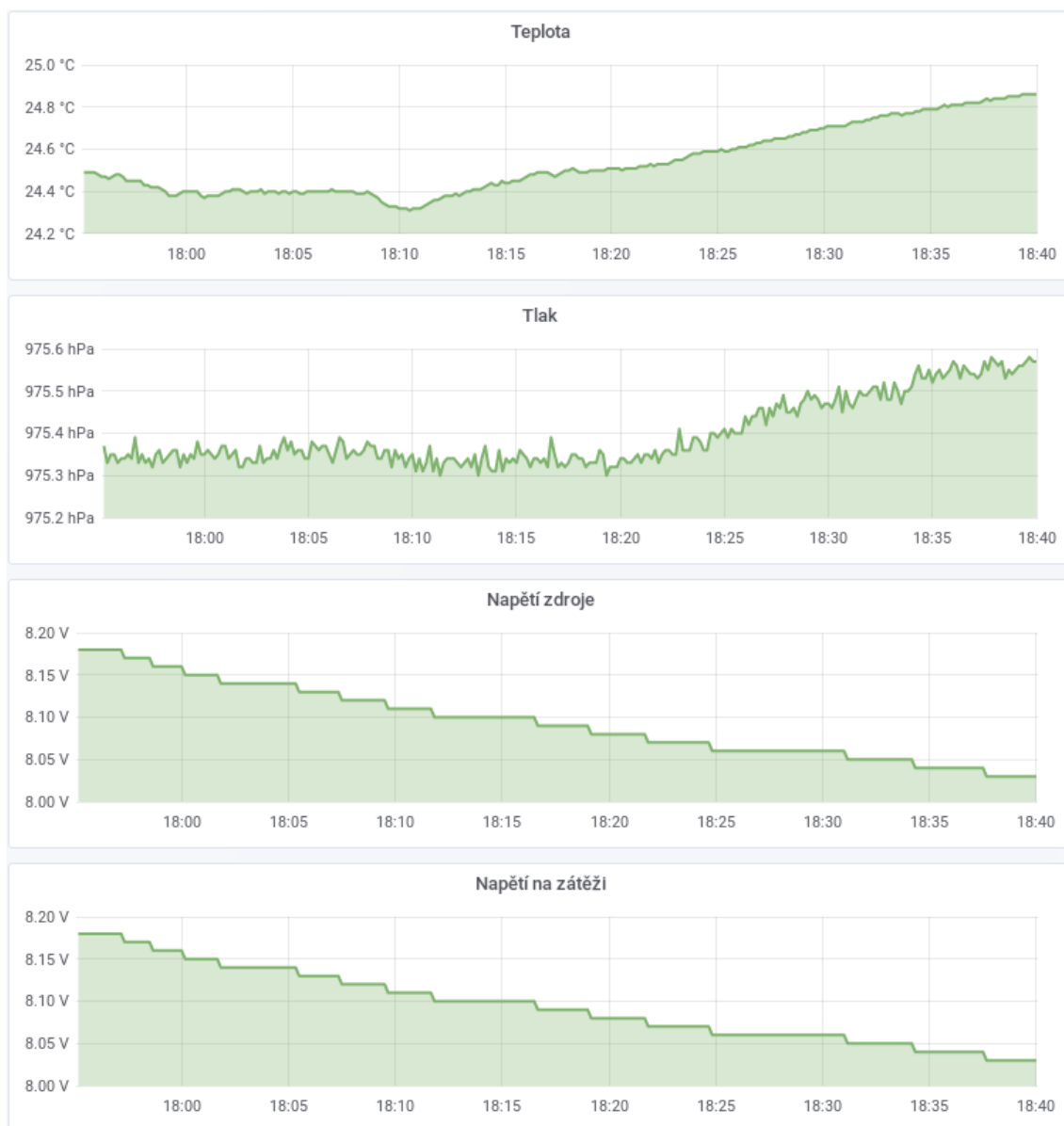
Nejprve je nutné nainstalovat software Grafana do našeho virtuálního počítače. Po úspěšné instalaci můžeme spustit Grafana server. Do Grafany se dostaneme přes webový prohlížeč na adrese <http://localhost:3000>. Port 3000 je výchozí http port, na kterém Grafana poslouchá.

Jakmile se dostaneme na tuto webovou stránku, je nejprve potřeba vytvořit nový datový zdroj, který bude Grafana využívat. V našem případě je to databáze InfluxDB s názvem senzory. Nyní, když máme přidáný datový zdroj, můžeme vytvořit novou pracovní plochu, kde můžeme vytvářet různé grafy pro zobrazení našich naměřených hodnot. Na obrázku 1.36 jsme si pomoci Grafana vytvořili například ukazatelé aktuální hodnoty teploty, vlhkosti, tlaku a napětí na zdroji.



Obrázek 1.36: Výpis aktuální hodnoty v Grafana

Na obrázcích 1.37 a 1.38 můžeme vidět ve vytvořených grafech jednotlivé průběhy měřených veličin v časovém intervalu 45 minut. Horizontální osa v grafu vždy udává čas a vertikální osa hodnotu měřené veličiny.



Obrázek 1.37: Vizualizace naměřených dat v Grafana část 1.



Obrázek 1.38: Vizualizace naměřených dat v Grafana část 2.

Závěr

Cílem bakalářské práce bylo zajistit sběr dat z různých senzorů, připojitelných k mikrokontroléru Arduino a jejich následný přenos do počítače s OS Linux. V teoretické části jsem popsal základní typy senzoru a používané komunikační protokoly pro přenos dat.

Hlavní náplň práce pak spočívala v praktické části, kde jsem navrhl způsob přenosu dat ze senzorů pomocí Bluetooth modulu připojeného k mikrokontroléru Arduino do virtuálního počítače s OS Linux. Díky tomu, že ve virtuálním počítači byly drobné problémy se sériovými porty, na které jsem posílal naměřená data, bylo s tím nutné při komunikaci přes USB i Bluetooth počítat. Menším problémem je také to, že data přicházejí na sériový port virtuálního počítače se zpožděním, což je nejspíše zapříčiněno tím, že přesměrováváme data z portu fyzického počítače na port ve virtuálním počítači. Data na port fyzického počítače s OS Windows totiž přicházejí bez viditelného zpoždění.

Samotný návrh tohoto řešení není nijak komplikovaný, strávil jsem však velké množství času nad tím, než jsem objevil, že lze celkem snadno a efektivně pomocí skriptu v jazyce Python číst data ze sériové portu, dále je zpracovávat a ukládat do databáze InfluxDB. Samotná databáze InfluxDB se velmi dobře osvědčila a je podle mého názoru velmi dobrou volbou pro ukládání dat v projektech podobného typu.

Výhodou tohoto návrhu je, že můžeme poměrně snadno připojit například další I2C senzory na vytvořenou sběrnici a provést pouze drobné rozšíření v kódu nahraném do mikrokontroléru Arduino. Také můžeme velmi snadno rozšířit používaný skript pro přijímání více naměřených hodnot.

Výsledky měření ukázaly, že vzhledem ke spotřebě tohoto vzniklého zařízení není příliš vhodné napájet Arduino pomocí 9 V baterie, ale raději jej napájet pomocí adaptéru z elektrické zásuvky nebo pomocí jiné alternativy, která vydrží napájet Arduino po delší dobu.

Použitá literatura

- [1] Arduino-shop.cz *Arduino měřič vzdálenosti ultrazvukový* [online]. [cit.2019-4-11]. Dostupné z: <https://arduino-shop.cz/arduino/846-arduino-meric-vzda-lenosti-ultrazvukovy.html>
- [2] Embedded Adventures *BME280* [online]. 2014 [cit.2019-4-11]. Dostupné z: <https://www.embeddedadventures.com/datasheets/BME280.pdf>
- [3] Laskarduino.cz *Doppler radar HB100, detektor pohybu* [online]. [cit.2019-4-11]. Dostupné z: https://laskarduino.cz/vstupni-periferie-cidla/230036-doppler-radar-hb100-detektor-pohybu.html?gclid=CjwKCAjwqfDlBRBDEiwAigXUaH5wfuQQpRZbCkNOn-Qf9OX9yk2fdLteEDtTJV2OzRf2aLdBagYBiBoC4WgQAvD_BwE
- [4] Arduino-shop.cz *I2C Modul pro rozpoznávání světla RGB pro Arduino* [online]. [cit.2019-4-11]. Dostupné z: <https://arduino-shop.cz/arduino/1712-i2c-modul-pro-rozpoznavani-svetla-rgb-pro-arduino.html>
- [5] Arduino-shop.cz *Detektor oxidu uhličitého CO2 MG811 s analogovým výstupem* [online]. [cit.2019-4-11]. Dostupné z: <https://arduino-shop.cz/arduino/2988-detektor-oxidu-uhliciteho-co2-mg811-s-analogovym-vystupem.html>
- [6] Arduino-shop.cz *Akcelerometr ADXL335 3-osy Analogový výstup* [online]. [cit.2019-4-11]. Dostupné z: <https://arduino-shop.cz/arduino/1260-akcelerometr-adxl335-3-osy-analogovy-vystup.html>
- [7] Karvinen T., Karvinen K., Valtokari V. *Make: Sensors: A Hands-On Primer for Monitoring the Real World with Arduino and Raspberry Pi*. Maker Media 2014 [cit.2019-04-11].
- [8] Circuit Basics *Basics of the SPI Communication Protocol* [online]. [cit.2019-4-11]. Dostupné z: <http://www.circuitbasics.com/basics-of-the-spi-communication-protocol>
- [9] Circuit Basics *Basics of UART Communication* [online]. [cit.2019-4-11]. Dostupné z: <http://www.circuitbasics.com/basics-uart-communication/>
- [10] Rahul Iyer *Arduino Serial Tutorial - Arduino Communication Protocols* [online]. 2016 [cit.2019-4-11]. Dostupné z: <https://www.deviceplus.com/how-tos/arduino-guide/arduino-communication-protocols-tutorial/>
- [11] Autor User: Cburnett *SPI timing diagram* [online]. 2006 [cit.2019-4-11]. Dostupné z: <https://commons.wikimedia.org/w/index.php?curid=1477421>
- [12] Olivka Petr, Seidl David *Computer Architectures and Paraller Systems* [online]. 2018 [cit.2019-4-11]. Dostupné z: <http://poli.cs.vsb.cz/edu/apps/lab/apps-labexc.pdf>
- [13] navody.arduino-shop.cz *Zjištění adresy I2C (TWI, wire) zařízení* [online]. [cit.2019-4-11]. Dostupné z: <https://navody.arduino-shop.cz/technikuv-blog/zjisteni-adresy-i2c-twi-wire-zarizeni.html>

- [14] Circuit Basics *Basics of the I2C Communication Protocol* [online]. [cit.2019-4-11]. Dostupné z: <http://www.circuitbasics.com/basics-uart-communication/>
- [15] Urbiš Hynek *Universal Serial Bus* [online]. 2000 [cit.2019-4-11]. Dostupné z: <http://www.fit.vutbr.cz/study/courses/PZ2/public/TEXTY/USB.pdf>
- [16] Autor neuveden *Universal Serial Bus* [online]. 2015 [cit.2019-4-11]. Dostupné z: https://cs.wikipedia.org/wiki/Universal_Serial_Bus
- [17] learn.sparkfun.com *Bluetooth Basics* [online]. [cit.2019-4-11]. Dostupné z: <https://learn.sparkfun.com/tutorials/bluetooth-basics/all>
- [18] Astrid Marcella, Karyono Kanisius *Communication from Smart Storage Container System using Bluetooth, Zigbee, and XML-RPC* [online]. 2011 [cit.2019-4-11]. Dostupné z: https://www.researchgate.net/figure/Piconet-Topology-12_fig7_313216468
- [19] Arduino-shop.cz *Arduino měření intenzity světla* [online]. [cit.2019-4-11]. Dostupné z: <https://arduino-shop.cz/arduino/902-arduino-mereni-intenzity-svetla.html>
- [20] Laskarduino.cz *Senzor tlaku, teploty a vlhkosti BME280* [online]. [cit.2019-4-11]. Dostupné z: https://laskarduino.cz/vstupni-periferie-cidla/131017-arduino-senzor-tlaku-teploty-a-vlhkosti-bme280.html?gclid=Cj0KCQjwtMvlBRDmARIsAEoQ8zQ6eImxTBrbhC2-qGUIJzfpM01QLURl04tt4iMTH4xr5sL_vbNP9swaAuLdEALw_wcB
- [21] Arduino-shop.cz *I2C senzor napětí a proudu 3kanálový INA3221* [online]. [cit.2019-4-11]. Dostupné z: <https://arduino-shop.cz/arduino/3080-i2c-senzor-napeti-a-proudu-3kanalovy-ina3221.html>
- [22] Arduino-shop.cz *Arduino Bluetooth modul HC-05* [online]. [cit.2019-4-11]. Dostupné z: <https://arduino-shop.cz/arduino/1005-arduino-bluetooth-modul-hc-05.html>
- [23] Autor neuveden *Proudový senzor ACS712* [online]. [cit.2019-4-11]. Dostupné z: <https://www.gme.cz/data/attachments/dsh.772-198.1.pdf>

Obsah elektronické přílohy

Příloha A:	<i>Knihovny pro senzor BH1750</i>
Příloha B:	<i>Kód pro senzor BH1750</i>
Příloha C:	<i>Knihovny pro senzor BME280</i>
Příloha D:	<i>Kód pro senzor BME280</i>
Příloha E:	<i>Knihovny pro senzor INA3221</i>
Příloha F:	<i>Kód pro senzor INA3221</i>
Příloha G:	<i>Kód pro testování Bluetooth modulu</i>
Příloha H:	<i>Celkový kód pro Arduino</i>
Příloha I:	<i>Skript pro ukládání dat</i>

